

目 录

一、	引言	3
二、	不可压缩的 Navier-Stokes 方程	5
三、	Navier-Stokes 方程的数值求解.....	9
四、	基于 POD/DEIM 方法的降阶模型.....	10
	(一)、 POD 和 DEIM 方法	10
	(二)、 POD/DEIM 方法在平流项和扩散项的应用	11
	(三)、 POD/DEIM 方法构造压力梯度项的降阶模型	14
五、	数值实验.....	16
六、	总结	18
	参考文献	18
	致谢	23

基于 POD/DEIM 方法的 Navier-Stokes 方程降阶模型

赖莹

南京信息工程大学数学与统计学院，江苏南京 210044

摘要：Navier-Stokes 方程是所有流体运动分析的核心，其中包括运动中的气体和等离子体。该方程常用于分析层流和湍流状态。本文提出了一种在交错网格下采用有限差分方法求二维不可压缩 Navier-Stokes 方程数值解的降阶模型，目的是节约计算成本。其思想是本征正交分解方法选取有限数量的 POD 基，并利用离散经验插值方法挑选出包含有效信息的插值值点来降低模型的维度，使得计算 Navier-Stokes 方程的整个过程在低维空间中进行。最后，在雷诺数 $Re = 100$ 的情况下，对比 POD-DEIM 降维模型和完整维度的数值模拟，证明了该降阶模型的可行性和有效性，在可接受的误差范围内实现了耗时减少到完整模型 1/10 的效果。

关键词：Navier-Stokes 方程；本征正交分解；离散经验插值；降阶模型

The Reduce-order Model of Navier-Stokes Equation Using POD/DEIM Method

Lai Ying

Abstract: The Navier-Stokes equation is typically at the heart of any analysis of fluid flow, which includes gases and plasma in motion. This equation is often employed to analyze both laminar and turbulent flow regimes. In this paper, we proposed a reduced order model for solving incompressible Navier Stokes equations using finite difference method on staggered grids. The idea is to select a finite number of POD bases using the proper orthogonal decomposition method (POD), and use the discrete empirical interpolation method (DEIM) to select points containing effective information to reduce the dimensionality of the model. This allows the entire process of calculating the Navier Stokes equation to be carried out in a low dimensional space. In the end of this paper, numerical simulations comparing the POD-DEIM reduced-dimension model with the full dimension for the Reynolds number $Re = 100$ demonstrate the feasibility and effectiveness of the reduced-dimension model, which achieves the effect of reducing the time consuming to 1/10 of the full model within an acceptable error range.

Keywords: Navier-Stokes Equation, Proper Orthogonal Decomposition, Discrete Empirical Interpolation Method, Reduced-Order Model

一、 引言

Navier-Stokes 方程是描述粘性流体物质运动的偏微分方程，其数学理论对于深入理解、预测和控制自然界和技术应用中的湍流具有根本的重要性。该方程在几十年间逐步建立理论得以发展，最早可以追溯到 Newton[1]的开创性工作，随后 Navier[2]、Poisson[3]和 St.Venant[4]等人进一步深入探索，最后经 Stokes[5]的研究该方程完整。Navier-Stokes 方程在数学上表达了牛顿流体的动量平衡和质量守恒。它们有时伴随着与压力、温度和密度相关的状态方程[6]。该方程与欧拉方程密切相关，两者之间的区别在于，前者考虑了粘性，而后者只模拟无粘流。因此，Navier-Stokes 方程是一个抛物线方程，从而具有更好的分析特性，但代价是数学结构较少(例如，它们永远不会完全可积)。Navier-Stokes 方程的应用也十分广泛，例如它们可用于非稳态的大气层建模[7]、洋流[8,9]、管道中的水流[10,11]和机翼和动物尾翼周围的气流[12,13]。完整形式和简化形式的 Navier-Stokes 方程有助于飞机和汽车的设计、血液流动的研究、发电站的设计、污染分析和许多其他事情。若结合 Maxwell 方程组，可模拟和研究磁流体动力学[14]

现阶段，关于 Navier-Stokes 方程的数值求解应用最广泛的是有限元方法[15-18]，这种方法通常是将大的域通过特定的空间离散化方法分解为更小、更简单的三角形单元网格，这些网格被称为“有限元”，每个小区域仅受空间上相邻的单元网格影响。其优点在于可以使用非结构化网格，这使得复杂几何形状的处理更加容易。此外，不发散的有限元方法从弱发散或完全不发散的自空间中逼近速度，使得压力项和不可压缩约束在鞍点系统中被消除，产生了一个对称正定系统，未知数大大减少[19]。值得注意的是在动量方程和连续性方程之间的相互作用下该方法会产生稳定性问题，通常需要利用速度-压力的“Taylor-Hood”结构[20]来构造基函数。

对比有限元方法能灵活应用在复杂边界的特点，有限差分通常规定只能在矩形域中操控，但是有限差分在编程仍有一定的优势，它更加简单，操作方便。更重要的是若基于分布在 Harlow 和 Welch[21]的交错网格，将速度项和压力项分散，使用有限差分方法能提高稳定性。据了解，Navier-Stokes 方程用有限差分求数值解的并不多见，但已有较多研究聚焦有限差分方法数值求解 Burgers 方程[22,23]，后者是前者去掉压力项的简化。Burgers 方程作为理解各种物理流动和问题的简单模型被广泛应用于各个领域，例如动力学建模、湍流现象，以及在粘性流体和交通流中传播的冲击波流动[24-26]。根据 Fletcher 的研究[27]，在多维空间中，使用有限差分的方案解 Burgers 方程相对有限元方法还有更经济的优点。

数值求解 Navier-Stokes 方程常用射影法，该方法最初由 Alexandre Chorin[28]提出，是一种快速有效的数值方法，用于解决非稳态 Navier-Stokes 方程。其基本思想是基于 Helmholtz-Hodge 分解，一个矢量场可以被唯一地分解为一个无发散的矢量场和一个无旋度的矢量场。射影法可分为两大类，即压力校正法和速度校正法，更多细节分别见[28-31]和[32-35]。后来，人们也根据基本的投射法，提出了一些新的方法。射影算法包括两个步骤。在第一步，计算不满足无发散条件的中间速度场。在第二步中，中间速度被分解为不发散的下一时间的速度和压力场。Kim 和 Moin[29]指出，对中间速度场使用传统的速度边界条件会导致不一致的数值解，他们因此为中间速度场导出了适当的边界条件。注意到求解 Navier-Stokes 方程的第一步可看作计算 Burgers 方程，二维 Burgers 方程相较不可压缩的 Navier-Stokes 方程，包含相同的平流项和扩散项，缺少压力梯

度项。受到启发，本文希望在已有的有限差分求解 Burgers 方程的基础上完成此方法对不可压缩 Navier-Stokes 方程的求解，即计算 Burgers 方程后得到速度场，再进一步处理压力梯度项，对速度进行修正。

此外，无论是有限元方法还是有限差分方法，两者都有昂贵计算成本的缺陷。在工程领域中，为了减轻计算分析的负担，人们寻找从整个数据源中提取有用信息的方法，即从高维数据下提取出相对更低维度的模式得到降阶模型。降阶模型的关键特征是它们能够从工程角度大幅降低数值计算，达到节约计算成本的目的。研究人员将降阶模型应用于电路设计[36]、固体力学中的多尺度建模[37]和金属成型过程[38]等多种工程问题。后来降阶模型在计算流体动力学中也被广泛应用，例如形状优化[39-42]，流量控制[43-45]和模拟计算流体力学问题上。现阶段，有关 Navier-Stokes 方程的降阶模型研究，人们开发了各种方法的降阶形式，例如 Domain decomposition[46]，有限元方法[18,47-49]，Petrov-Galerkin Methods[50]，Finite Volume Methods[51-53]，Residual DEIM method[54]和 RBF interpolation[55]等方法。以上方法大部分都利用了一种有效的降阶技术，即结合 Galerkin 投影的本征正交分解(Proper Orthogonal Decomposition，简称 POD)。POD 方法于 1901 年引入，当时被 Pearson 称为主成分分析(PCA)[56]。POD 已被证明是包括湍流建模在内的一些科学领域的有效方法，例如[57]和[58]。在 POD 降阶建模中，Galerkin 投影方法对完整的 Navier-Stokes 方程进行离散，通过 snapshot 方法将原始方程投影到有限数量的 POD 基上，并得到一组时间上的常微分方程，从而达到降低维度的目的。除了效率外，POD-Galerkin 方法仍存在一些缺点。Chaturantabut[59-60]在她的硕士和博士论文中首先提出了一种新的方法来解决二次非线性项的问题，被称为预算算 POD 技术。它包括利用二次非线性的结构，通过操控计算顺序和分离空间变量与时间变量，来消除对原始离散系统维度的依赖。但是对于高阶非线性项，非线性项必须回到原维度中计算，使得计算成本增加，这时 POD/离散经验插值法(Discrete Empirical Interpolation Method，简称 POD DEIM)是首选的方法。DEIM 是 Barrault 等人提出的经验插值法(EIM)[61]的一个离散变体，用于构造非仿射参数化函数的近似，该方法是在降阶模型的背景下提出的。DEIM 方法采用了一小部分选定的网格点，以避免在非线性项所需的每个时间步长上计算内积造成的时间冗余。DEIM 方法已被证明能够在 CPU 时间上比原始非还原模型快 10-100 倍。CPU 时间的节约与降阶模型的维度成正比(见 Stefanescu 和 Navon[62])，因此与网格点的数量成正比。Chaturantabut 和 Sorensen[59,60,63]提出并分析了在离散经验插值法(DEIM)框架下应用 POD。对经验插值法(EIM)的其他有关降阶方法的重要贡献包括 Barrault 等人[61]和 Patera[64-69]。在实际应用中，POD/DEIM 的降阶方法是对非线性部分模拟近似，只需要非线性项的几个分量。利用这几个分量，其余的分量通过插值进行近似还原出来。与其他能够减少方程组中的非线性项维度的方法相比，DEIM 的优势在于它能够从完整维度模型中自动选择其分量或匹配点。这样做的好处是用户不需要手动输入。

本文的目标是在有限差分方法求解不可压缩的 Navier-Stokes 方程的情况下，利用 POD/DEIM 方法构建对应的降阶模型。Wang[70]在论文中证实了有限差分方法下基于 Galerkin 投影的 POD/DEIM 的降阶模型在二维 Burgers 方程上的可行性。基于此本文首先在忽略压力项的情况下，先对有限差分形式的平流项和扩散项进行降阶处理后得到低维的中间速度场，再在已有低维速度场下对压力项构建降阶模型，从而使得整个 Navier-Stokes 的数值计算始终保持在低维空间中进行。这样做能提高计算速度，节约计算成本。

本文的其他内容分布如下。第二节中给出了不可压缩的二维 Navier-Stokes 方程形式及其在交错网格下的有限差分离散化公式。第三节展示了 Navier-Stokes 方程的数值求解步骤。第四节简单总结了矩阵-向量形式的 Navier-Stokes 方程并详细描述了采用 POD/DEIM 方法的降阶模型的构建过程。第五节进行了数值实验来验证模型的性能。本文的最后提供了一个结论性的建议，并在附录补充了 DEIM 算法和完整模型和降阶模型的时间复杂度计算。

二、不可压缩的 Navier-Stokes 方程

在二维空间内，随时间变化的不可压缩 Navier-Stokes 方程公式如下：

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = \frac{1}{Re} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) - \frac{\partial p}{\partial x} \quad (1a)$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = \frac{1}{Re} \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) - \frac{\partial p}{\partial y} \quad (1b)$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0. \quad (1c)$$

其中 Re 表示流动惯性的比例与流动中粘性力的比例， $(u(x, y, t), v(x, y, t))$ 代表流场， $p(x, y, t)$ 代表压力。该方程的推导见[71]。公式(1c)表示不可压缩流体[72]。设定此方程在矩形域 $\Omega = [0, l_x] \times [0, l_y]$ 上，该区域在时间上是固定的，再规定四个边界分别表示为北(N)、南(S)、西(W)和东(E)，设定每个边界上的边界条件如下

$$\begin{aligned} u(x, l_y, t) &= u_N(x, t) & v(x, l_y, t) &= 0 & u(x, 0, t) &= u_S(x, t) & v(x, 0, t) &= 0 \\ v(0, y, t) &= v_W(y, t) & u(0, y, t) &= 0 & v(l_x, y, t) &= v_E(y, t) & u(l_x, y, t) &= 0 \end{aligned} \quad (2)$$

因为 Navier-Stokes 方程满足 $\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$ ，故有

$$\begin{aligned} \frac{\partial u^2}{\partial x} + \frac{\partial(u, v)}{\partial y} &= 2u \frac{\partial u}{\partial x} + u \frac{\partial v}{\partial y} + v \frac{\partial u}{\partial y} = u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} \\ \frac{\partial v^2}{\partial y} + \frac{\partial(u, v)}{\partial x} &= 2v \frac{\partial v}{\partial y} + u \frac{\partial v}{\partial x} + v \frac{\partial u}{\partial x} = u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y}. \end{aligned} \quad (3)$$

从而得到守恒形式的二维不可压缩 Navier-Stokes 方程：

$$\begin{aligned} \frac{\partial u}{\partial t} + \frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y} &= \frac{1}{Re} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) - \frac{\partial p}{\partial x} \\ \frac{\partial v}{\partial t} + \frac{\partial uv}{\partial x} + \frac{\partial v^2}{\partial y} &= \frac{1}{Re} \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) - \frac{\partial p}{\partial y} \\ \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} &= 0. \end{aligned} \quad (4)$$

这种形式的 Navier-Stokes 方程在数值处理上更容易，后文的微分算子近似都将基于公式(4)描述。

为了研究 Navier-Stokes 方程的流场，要引入流函数 q 。流函数的引入能直观反映流体的运动状态，相比速度场更加直观。流函数 q 的正交梯度是速度场：

$$(\nabla q)^\perp = \mathbf{u} \Leftrightarrow -q_y = u \text{ and } q_x = v.$$

将二维旋度应用于此等式，得到：

$$-\Delta q = \nabla \times (\nabla q)^\perp = \nabla \times \mathbf{u} \Leftrightarrow -\Delta q = -q_{yy} - q_{xx} = u_y - v_x,$$

这个流函数仅当 $\Delta \cdot \mathbf{u} = u_x + v_y = 0$ 时成立。

表 1: 不同场的大小

场	内点数	含边界的点数
压力场 p	$n_x \times n_y$	$(n_x + 2) \times (n_y + 2)$
速度场 u	$(n_x - 1) \times n_y$	$(n_x + 1) \times (n_y + 2)$
速度场 v	$n_x \times (n_y - 1)$	$(n_x + 2) \times (n_y + 1)$

利用笛卡尔几何学, 计算域 Ω 被划分为多个均匀的网格, 网格间距固定。本文采用 Harlow 和 Welch[21]的交错标记-单元(Marker and Cell, MAC)网格, 每个单元格的中心代表压力项 p , 速度 u 被放置在单元格垂直边界上, 速度 v 被放置在单元格水平边界上, 这样做有利于提高稳定性。图 1 展示了 $n_x = 4, n_y = 3$ 的交错网格示意图。此外必须注意参数 p, u, v 的内部点和边界点。本文规定域内的任意点都是内点, 而边界上和边界外的点都是虚点。图 1 中的深色标记代表内部点, 而浅色标记代表边界点。这些场的大小如表 1 所示:

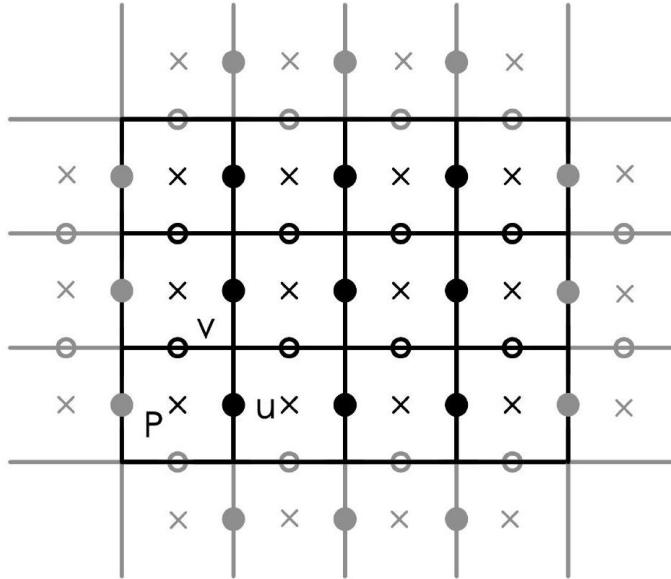


图 1: 交错网格示意图

当二维空间计算域 Ω 在 x (水平), y (垂直)方向分别均匀地划分为 $n_x - 1$ 和 $n_y - 1$ 个子区间, 假设离散函数定义在空间域 $\Omega = [0, l_x] \times [0, l_y]$ 的 $n_x \times n_y$ 网格上。单元格的中心位于 $(x_i, y_j) = ((i - 0.5)h, (j - 0.5)h), i = 1, \dots, n_x, j = 1, \dots, n_y$, 单元格顶点位于 $(x_i + \frac{1}{2}, y_j + \frac{1}{2}) = (ih, jh)$ 。下面会用到这些定义: $x_i = i\Delta x, y_j = j\Delta y, u_{ij} = u(x_i, y_j, t), v_{ij} = v(x_i, y_j, t)$, 其中 $\Delta x = \frac{l_x}{n_x - 1}, \Delta y = \frac{l_y}{n_y - 1}$ 。

首先在时间上, 以 u 为满足:

$$\left(\frac{\partial u}{\partial t} \right)_{i+\frac{1}{2},j}^{n+1} = \frac{u_{i+\frac{1}{2},j}^{n+1} - u_{i+\frac{1}{2},j}^n}{\Delta t}, \quad (5)$$

同理, 可以得到 v 在时间上的计算公式。

对于二阶差分, 通过利用中心差分思想来求拉普拉斯算子, 对于 $u_{i+\frac{1}{2},j}$ 有:

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_{i+\frac{1}{2},j}^{n+1} = \frac{u_{i+\frac{3}{2},j}^{n+1} - 2u_{i+\frac{1}{2},j}^{n+1} + u_{i-\frac{1}{2},j}^n}{\Delta x^2}. \quad (6)$$

该公式在实际应用中会存在使用边界点的情况。相同的公式也适用于分量 v 以及压力项 p 的拉普拉斯算子近似。

对于一阶差分，同样利用中心差分的思想求，下面分别给出 $p_{i+\frac{1}{2},j}$ 和 $u_{i+\frac{1}{2},j}^2$ 的一阶差分算子近似：

$$\begin{aligned} \left(\frac{\partial p}{\partial x}\right)_{i+\frac{1}{2},j}^{n+1} &= \frac{p_{i+1,j}^{n+1} - p_{i,j}^{n+1}}{\Delta x}, \\ \left(\frac{\partial u^2}{\partial x}\right)_{i+\frac{1}{2},j}^n &= \frac{(u_{i+1,j}^2 - u_{i,j}^2)^n}{\Delta x}. \end{aligned} \quad (7)$$

其中通过对两个相邻值进行插值来获得 $u_{i+1,j}, u_{i,j}$ ：

$$\begin{cases} u_{i+1,j} &= \frac{1}{2}(u_{i+\frac{1}{2},j} + u_{i+\frac{3}{2},j}) \\ u_{i,j} &= \frac{1}{2}(u_{i+\frac{1}{2},j} + u_{i-\frac{1}{2},j}) \end{cases}$$

对于交叉项，满足：

$$\begin{aligned} \left(\frac{\partial uv}{\partial y}\right)_{i+\frac{1}{2},j}^n &= \frac{1}{\Delta y} \left[(uv)_{i+\frac{1}{2},j+\frac{1}{2}}^n - (uv)_{i+\frac{1}{2},j-\frac{1}{2}}^n \right] \\ &= \frac{1}{\Delta y} \left[u_{i+\frac{1}{2},j+\frac{1}{2}}^n v_{i+\frac{1}{2},j+\frac{1}{2}}^n - u_{i+\frac{1}{2},j-\frac{1}{2}}^n v_{i+\frac{1}{2},j-\frac{1}{2}}^n \right], \end{aligned}$$

其中：

$$\begin{cases} u_{i+\frac{1}{2},j+\frac{1}{2}} &= \frac{1}{2}(u_{i+\frac{1}{2},j} + u_{i+\frac{1}{2},j+1}) \\ u_{i+\frac{1}{2},j-\frac{1}{2}} &= \frac{1}{2}(u_{i+\frac{1}{2},j-1} + u_{i+\frac{1}{2},j}) \\ v_{i+\frac{1}{2},j+\frac{1}{2}} &= \frac{1}{2}(v_{i,j+\frac{1}{2}} + v_{i+1,j+\frac{1}{2}}) \\ v_{i+\frac{1}{2},j-\frac{1}{2}} &= \frac{1}{2}(v_{i,j-\frac{1}{2}} + v_{i+1,j-\frac{1}{2}}) \end{cases}$$

此外，边界条件的选择应当谨慎。本文采用盖驱动空腔问题的边界条件来后续验证计算方法的可行性，这是一个关于粘性不可压缩流体的 Benchmark 问题[73]，虽然边界条件相对简单，但创建的流动特征却非常有趣和复杂。盖驱动空腔包含了 3 个无滑移刚性壁和一个切向移动的盖子，左下角的压力设定为 0，示意图见图 2。本文规定速度场 u 和 v 的边界条件为狄利克雷边界，压力场 p 则为诺依曼边界。边界条件具体如何应用也需要具体位置具体分析，因为对于交错网格来说，有些点位于边界上，而有些点之间存在一个边界。在位于边界上的点，其数值是直接确定的，例如 u 在西边和东边的边界， v 在北边和南边的边界。而对于南北边界的速度 u 和东西边界的速度 v ，则取相邻两点的平均值。设边界以下分别位于边界上方的两点为 $u_{i,j}$ 和 $u_{i,j+1}$ ，规定的边界值为 u_N 。那么边界条件是

$$\frac{u_{i,j} + u_{i,j+1}}{2} = u_N \Leftrightarrow u_{i,j} + u_{i,j+1} = 2u_N. \quad (8)$$

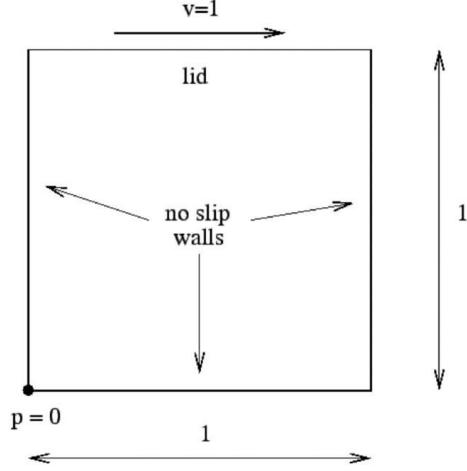


图 2: 盖驱动空腔示意图

类似得到在南边界的 u_S 和在西东边界的 v_W, v_E 。

边界处的法向导数 $\frac{\partial p}{\partial n}$ 是由边界内外两个点定义的。例如，在北边界，规定齐次诺依曼边界条件满足：

$$\frac{p_{i,j+1} - p_{i,j}}{h_y} = 0 \Leftrightarrow p_{i,j+1} = p_{i,j}. \quad (9)$$

最终得到在该空间的一阶、二阶导数的中心差分形式的 Navier-Stokes 方程如下：

$$\begin{aligned} & \frac{du_{i+\frac{1}{2},j}}{dt} + \frac{1}{4\Delta x} \left[\left(u_{i+\frac{3}{2},j} + u_{i+\frac{1}{2},j} \right)^2 - \left(u_{i+\frac{1}{2},j} + u_{i-\frac{1}{2},j} \right)^2 \right] \\ & + \frac{1}{4\Delta y} \left[\left(u_{i+\frac{1}{2},j+1} + u_{i+\frac{1}{2},j} \right) \left(v_{i,j+\frac{1}{2}} + v_{i+1,j+\frac{1}{2}} \right) - \left(u_{i+\frac{1}{2},j} + u_{i+\frac{1}{2},j-1} \right) \left(v_{i,j-\frac{1}{2}} + v_{i+1,j-\frac{1}{2}} \right) \right] \\ & = \frac{1}{Re \Delta x^2} \left(u_{i+\frac{3}{2},j} - 2u_{i+\frac{1}{2},j} + u_{i-\frac{1}{2},j} \right) + \frac{1}{Re \Delta y^2} \left(u_{i+\frac{1}{2},j+1} - 2u_{i+\frac{1}{2},j} + u_{i+\frac{1}{2},j-1} \right) - \frac{p_{i+1,j} - p_{i,j}}{\Delta x}. \quad (10) \\ & \frac{dv_{i,j+\frac{1}{2}}}{dt} + \frac{1}{4\Delta y} \left[\left(v_{i,j+\frac{3}{2}} + v_{i,j+\frac{1}{2}} \right)^2 - \left(v_{i,j+\frac{1}{2}} + v_{i,j-\frac{1}{2}} \right)^2 \right] \\ & + \frac{1}{4\Delta x} \left[\left(v_{i+1,j+\frac{1}{2}} + v_{i,j+\frac{1}{2}} \right) \left(u_{i+\frac{1}{2},j} + u_{i+\frac{1}{2},j+1} \right) - \left(v_{i,j+\frac{1}{2}} + v_{i-1,j+\frac{1}{2}} \right) \left(u_{i-\frac{1}{2},j} + u_{i-\frac{1}{2},j+1} \right) \right] \\ & = \frac{1}{Re \Delta x^2} \left(v_{i+1,j+\frac{1}{2}} - 2v_{i,j+\frac{1}{2}} + v_{i-1,j+\frac{1}{2}} \right) + \frac{1}{Re \Delta y^2} \left(v_{i,j+\frac{3}{2}} - 2v_{i,j+\frac{1}{2}} + v_{i,j-\frac{1}{2}} \right) - \frac{p_{i+1,j} - p_{i,j}}{\Delta y}. \end{aligned}$$

如果流体运动速度并不快，上述的中心差分形式是合适的，但是对于更快的流或更大的时间步长，离散形式应该更接近迎风格式。根据 Benjamin Seibold 的研究[74]，本文使用参数 $\gamma \in [0,1]$ 在中心差分和迎风格式之间实现相互平稳过渡， γ 定义为：

$$\gamma = \min \left(1.2 \cdot \Delta t \cdot \max \left(\max_{i,j} \left| u_{i+\frac{1}{2},j} \right|, \max_{i,j} \left| v_{i,j+\frac{1}{2}} \right| \right), 1 \right), \quad (11)$$

这里的 1.2 是根据经验得出的，为了准确性，通常更倾向于逆风可能是有利的[75]。中心差分和逆风之间的线性组合是通过以下方式实现的，不妨设 h, v 的上标分别代表水平和竖直方向，平均值和差分量的计算方式如下：

$$\begin{aligned} \bar{u}_{i,j}^h &= \frac{u_{i-\frac{1}{2},j} + u_{i+\frac{1}{2},j}}{2}; \quad \bar{u}_{i,j}^v = \frac{u_{i,j-\frac{1}{2}} + u_{i,j+\frac{1}{2}}}{2}; \\ \check{u}_{i,j}^h &= \frac{u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}}{2}; \quad \check{u}_{i,j}^v = \frac{u_{i,j+\frac{1}{2}} - u_{i,j-\frac{1}{2}}}{2}. \end{aligned} \quad (12)$$

这种思路同样适用于 $\bar{v}^h, \bar{v}^v, \check{v}^h, \check{v}^v$ 。有了以上的定义，Navier-Stokes 方程的非线性项，即平流项

的有限差分形式计算公式更改为:

$$\begin{aligned} ((u^n)^2)_x + (u^n v^n)_y &= ((\bar{u}^h)^2 - \gamma |\bar{u}^h| \check{u}^h)_x + (\bar{u}^v \bar{v}^h - \gamma |\bar{v}^h| \check{u}^v)_y, \\ (u^n v^n)_x + ((v^n)^2)_y &= (\bar{u}^v \bar{v}^h - \gamma |\bar{u}^v| \check{v}^h)_x + ((\bar{v}^v)^2 - \gamma |\bar{v}^v| \check{v}^v)_y. \end{aligned} \quad (13)$$

因此对于 u^2 来说, 其近似值可以用如下公式计算:

$$\begin{aligned} ((\bar{u}^h)^2 - \gamma |\bar{u}^h| \check{u}^h)_{i,j} &= |\bar{u}^h| (|\bar{u}^h| - \gamma \check{u}^h)_{i,j} \\ &= \bar{u}^h \begin{cases} \left(\frac{1-\gamma}{2}\right) u_{i+\frac{1}{2},j} + \left(\frac{1+\gamma}{2}\right) u_{i-\frac{1}{2},j} & \text{if } \bar{u}^h \geq 0 \\ \left(\frac{1+\gamma}{2}\right) u_{i+\frac{1}{2},j} + \left(\frac{1-\gamma}{2}\right) u_{i-\frac{1}{2},j} & \text{if } \bar{u}^h < 0 \end{cases} \end{aligned} \quad (14)$$

可以很容易发现当采用中心差分的时候 $\gamma = 0$, 而使用守恒形式的迎风格式时 $\gamma = 1$ 。

三、 Navier-Stokes 方程的数值求解

根据投射法, 不可压缩的 Navier-Stokes 方程的数值计算主要分为两个步骤, 首先忽视压力梯度项计算动量方程, 求得一个中间速度。然后再利用压力项对中间速度进行更正得到最终的速度场, 下面给出全维度 Navier-Stokes 方程的数值计算模型的详细求解步骤, 因篇幅限制, 在此处不给出有限差分格式的公式, 仍以原方程为基础描述。

在假设已知流速场 u^n, v^n 的情况下, 求下一个时间步长的 u^{n+1}, v^{n+1} , 投射法的第一步处理非线性的平流项(15)和扩散项(16):

$$\begin{cases} \frac{u^* - u^n}{\Delta t} = -((u^n)^2)_x - (u^n v^n)_y \\ \frac{v^* - v^n}{\Delta t} = -(u^n v^n)_x - ((v^n)^2)_y \end{cases} \quad (15)$$

$$\begin{cases} \frac{u^{**} - u^*}{\Delta t} = \frac{1}{Re} (u_{xx}^{**} + u_{yy}^{**}) \\ \frac{v^{**} - v^*}{\Delta t} = \frac{1}{Re} (v_{xx}^{**} + v_{yy}^{**}) \end{cases} \quad (16)$$

注意到利用公式(15), (16), 能够消去 u^*, v^* , 得到:

$$\begin{cases} \frac{u^{**} - u^n}{\Delta t} = -((u^n)^2)_x - (u^n v^n)_y + \frac{1}{Re} (u_{xx}^{**} + u_{yy}^{**}) \\ \frac{v^{**} - v^n}{\Delta t} = -(u^n v^n)_x - ((v^n)^2)_y + \frac{1}{Re} (u_{xx}^{**} + u_{yy}^{**}) \end{cases} \quad (17)$$

上述公式(17)的右边部分仅含有平流项和对流项, 因为不可压缩流体条件, 这一步骤的处理相当于解 Burgers 方程, 即 u^{**}, v^{**} 是 Burgers 方程的解, 在本文中被视作称为中间速度。

根据 Navier-Stokes 方程, u^{n+1}, v^{n+1} 和 u^{**}, v^{**} 只相差压力梯度项, 故有:

$$\begin{cases} \frac{u^{n+1} - u^{**}}{\Delta t} = -\left(\frac{\partial p}{\partial x}\right)^{n+1} \\ \frac{v^{n+1} - v^{**}}{\Delta t} = -\left(\frac{\partial p}{\partial y}\right)^{n+1} \end{cases} \quad (18)$$

不妨设 $\vec{V} = (u, v)^T$ 。在时间 $n+1$ 上 \vec{V} 满足 $\nabla \cdot \vec{V}^{n+1} = 0$, 从而得出椭圆压力公式:

$$\nabla \cdot \vec{V}^{n+1} = \nabla \cdot (\nabla p)^{n+1} = \Delta p^{n+1}. \quad (19)$$

完整的压力项校正步骤主要分为四步。在第一步解决 Burgers 方程后，首先可以很容易计算出 $F^n = \nabla \cdot \vec{V}^{**}$ ，接下来再解泊松方程

$$\Delta p^{n+1} = \frac{1}{\Delta t} F^n, \quad (20)$$

得到压力 p 值后，进一步计算压力项的一元微分算子结果 $G^{n+1} = \nabla p^{n+1}$ ，最终得到对速度场更新，这样就完成了一个时间步长内的速度更新。

$$\vec{V}^{n+1} = \vec{V}^{**} - \Delta t G^{n+1}. \quad (21)$$

四、基于 POD/DEIM 方法的降阶模型

(一) POD 和 DEIM 方法

本征正交分解 (POD) 是一种在希尔伯特空间中构造子空间的低维近似表示的方法。它本质上与有限维空间或欧几里得空间中的奇异值(Singular Value Decomposition, 简写为 SVD)分解相同。它有效地提取了包含偏微分方程解空间特征的基本元素。欧几里得空间中的 POD 基定义如下：

定义 1：给定一组 snapshots : $\{\mathbf{y}^1, \dots, \mathbf{y}^{n_s}\} \subset \mathbb{R}^n$ (snapshots 是轨迹的样本)，令 $\gamma = \text{span}\{\mathbf{y}^1, \dots, \mathbf{y}^{n_s}\} \subset R^n$ 和 $r = \dim(\gamma)$ 。维度 $k < r$ 的 POD 基是一组正交向量集合 $\{\Phi\}_{i=1}^k \subset R^n$ ，其线性张成空间最近似空间 γ 。基组成的集合 $\{\Phi\}_{i=1}^k$ 解决了最小化问题

$$\min_{\{\Phi_i\}_{i=1}^k} \sum_{j=1}^{n_s} \left\| \mathbf{y}_j - \sum_{i=1}^k (\mathbf{y}_j^T \Phi) \Phi \right\|_2^2, \quad (22)$$

其中

$$\Phi_i^T \Phi_j = \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \quad (23)$$

公式(22)的解是 snapshots 矩阵 $\mathbf{Y} = [\mathbf{y}^1, \dots, \mathbf{y}^{n_s}] \in \mathbb{R}^{n \times n_s}$ 的左奇异向量集。特别地，假设 \mathbf{Y} 的 SVD 分解是

$$\mathbf{Y} = \mathbf{V} \Sigma \mathbf{W}^T, \quad (24)$$

其中 $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_r] \in \mathbb{R}^{n \times r}$, $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_r] \in \mathbb{R}^{n_s \times r}$ 是正交的, $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_r) \in \mathbb{R}^{r \times r}$, 对角线元素满足 $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$, \mathbf{Y} 的秩为 r , 且有 $r \leq \min(n, n_s)$ 。如此一来公式(22)的 POD 基或最优解是 $\{\mathbf{v}_i\}_{i=1}^k$ 。从而 POD 基下近似 snapshots 的最小二范数误差为：

$$\sum_{j=1}^{n_s} \left\| \mathbf{y}_j - \sum_{i=1}^k (\mathbf{y}_j^T \mathbf{v}_i) \mathbf{v}_i \right\|_2^2 = \sum_{i=k+1}^r \sigma_i^2. \quad (25)$$

对于 k 值的选取，实际应用中多采用能量占比的方法来决定，即通过公式(26)来决定。通常会选择能满足 $I(k) > 80\%$ 或 90% 的 k 值。

$$I(k) = \frac{\sum_{i=1}^k \sigma_i^2}{\sum_{i=1}^n \sigma_i^2}. \quad (26)$$

仅用 POD-Galerkin 方法获得的降阶模型仍有低效率的不足，甚至在某些情况下，降阶模型

的计算成本很可能和原维度的运算成本一致[60]。为了解决这个问题, Chaturantabut[59]提出将非线性函数投射到一个子空间上, 该子空间近似于非线性函数产生的空间, 并且是由维数为 $m \ll n$ 的基张成得到的空间。假设已有非线性方程 $f(t)$, 将 $f(t)$ 投影到基 $\{u_1, \dots, u_m\} \in \mathbb{R}^n$ 上, 将在这组基上对应系数向量记为 $\mathbf{c}(\tau)$, 有

$$\mathbf{f}(\tau) \approx \mathbf{U}\mathbf{c}(\tau), \quad (27)$$

接下来考虑降维 (选取行元素) 矩阵

$$\mathbf{P} = [\mathbf{e}_{\phi_1}, \dots, \mathbf{e}_{\phi_m}] \in \mathbb{R}^{n \times m}, \quad (28)$$

其中, $\mathbf{e}_{\phi_i} = [0, \dots, 0, 1, 0, \dots, 0]^T \in \mathbb{R}^n$ 是单位矩阵 $\mathbf{I}_n = \mathbb{R}^{n \times n}$ 的第 ϕ_i 列, $i = 1, \dots, m$ 。假设 $\mathbf{P}^T \mathbf{U}$ 是非奇异的, 通过矩阵 \mathbf{P}^T 实现了对SVD正交基 \mathbf{U} 的超定矩阵降维。

$$\mathbf{P}^T \mathbf{f}(\tau) = (\mathbf{P}^T \mathbf{U}) \mathbf{c}(\tau), \quad (29)$$

并且将公式(27)化为最终的近似形式:

$$\mathbf{f}(\tau) \approx \mathbf{U}\mathbf{c}(\tau) = \mathbf{U}(\mathbf{P}^T \mathbf{U})^{-1} \mathbf{P}^T \mathbf{f}(\tau). \quad (30)$$

为了获得这个近似值, 需要明确基 $\{u_1, u_2, \dots, u_m\}$ 和在(28)中用到的插值指标 $\{\phi_1, \phi_2, \dots, \phi_m\}$ 。其中非线性的投影基 $\{u_1, u_2, \dots, u_m\}$ 是用 POD 方法处理原全阶系统中得到的非线性 snapshots, 从而构建非线性函数 \mathbf{f} 的投影基。插值指标 $\{\phi_1, \phi_2, \dots, \phi_m\}$, 用于确定近似方程(27)中的 $\mathbf{c}(\tau)$, 将通过基于贪婪搜索思想的 DEIM 算法得到 (见附录A)。

(二)、POD/DEIM 方法在平流项和扩散项的应用

为了对方程中的变量降阶, 首先将二维空间的速度项 u, v 以及压力项 p 展开成向量, 并将相应的数值差分算子化为矩阵。这样的思路是源于 Wang 对二维 Burgers 方程的 POD/DEIM 降阶模型[70]。

在交错网格格式中, 为了后续降阶和编程, 本文将向量 U 和 V 在矩阵中按不同的方向展开, 其中 U 按 x 方向展开, V, P 按 y 方向展开。设向量 U, V 和 P 分别为:

$$\begin{aligned} U &= \left(u_{\frac{3}{2},1}, u_{\frac{5}{2},1}, \dots, u_{n+\frac{1}{2},1}, u_{\frac{3}{2},2}, \dots, u_{n+\frac{1}{2},2}, \dots, u_{\frac{3}{2},n}, \dots, u_{n+\frac{1}{2},n} \right)^T, \\ V &= \left(v_{1,\frac{3}{2}}, v_{1,\frac{5}{2}}, \dots, v_{1,n+\frac{1}{2}}, v_{2,\frac{3}{2}}, \dots, v_{2,n+\frac{1}{2}}, \dots, v_{n+\frac{1}{2},1}, \dots, v_{n,n+\frac{1}{2}} \right)^T \\ P &= (p_{11}, p_{12}, \dots, p_{1n}, p_{21}, \dots, p_{2n}, \dots, p_{n1}, \dots, p_{nn})^T. \end{aligned} \quad (31)$$

因此, 守恒形式的不可压缩 Navier-Stokes 方程在有限差分下对应的矩阵-向量形式为:

$$\begin{aligned} \frac{dU}{dt} + f_1(U, V) &= H_u(U + k_x \tilde{U} + k_y \hat{U}) - \frac{\partial P}{\partial x}. \\ \frac{dV}{dt} + f_2(U, V) &= H_v(V + k_y \tilde{V} + k_x \hat{V}) - \frac{\partial P}{\partial y}. \end{aligned} \quad (32)$$

其中 $k_x = t/Re/\Delta x^2, k_y = dt/Re/\Delta y^2$ 。 \tilde{U}, \tilde{V} 表示 U, V 的边界点向量; f_1, f_2 代表 Navier-Stokes 方程中的平流项。 \hat{U}, \hat{V} 则是 U, V 的虚点(边界外)向量; H_u, H_v 为五对角二阶差分矩阵, 它们在整个时间步长计算中不会改变。

两个平流项投影 $f_1, f_2: \mathbb{R}^{n_{xy}} \times \mathbb{R}^{n_{xy}} \rightarrow \mathbb{R}^{n_{xy}}$ 的具体形式如下:

$$\begin{aligned}
f_1(U, V) = & \frac{1}{4\Delta x} \left\{ \left[(M_{Ru}U + \tilde{U}_R)^2 - (M_{Lu}U + \tilde{U}_L)^2 \right] \right. \\
& - \gamma \left[|M_{Ru}U + \tilde{U}_R|.* (M_{Rut}U + \tilde{U}_R) - |M_{Lu}U + \tilde{U}_L|.* (M_{Lut}U - \tilde{U}_L) \right] \} \\
& - \frac{1}{4\Delta y} \left\{ \left[(\Theta_{Ru}U + \hat{U}_R).* (\Xi_{Rv}V + d_{v1}) - (\Theta_{Lu}U + \hat{U}_L).* (\Xi_{Lv}V + d_{v2}) \right] \right. \\
& - \gamma \left[(\Theta_{Rut}U + \hat{U}_R).* |\Xi_{Rv}V + d_{v1}| - (\Theta_{Lut}U + \hat{U}_L).* |\Xi_{Lv}V + d_{v2}| \right] \}, \quad (33) \\
f_2(U, V) = & \frac{1}{4\Delta y} \left\{ \left[(M_{Rv}V + \tilde{V}_R)^2 - (M_{Lv}V + \tilde{V}_L)^2 \right] \right. \\
& - \gamma \left[|M_{Rv}V + \tilde{V}_R|.* (M_{Rvt}V + \tilde{V}_R) - |M_{Lv}V + \tilde{V}_L|.* (M_{Lvt}V - \tilde{V}_L) \right] \} \\
& - \frac{1}{4\Delta x} \left\{ \left[(\Theta_{Rv}V + \hat{V}_R).* (\Xi_{Ru}U + d_{u1}) - (\Theta_{Lv}V + \hat{V}_L).* (\Xi_{Lu}U + d_{u2}) \right] \right. \\
& - \gamma \left[(\Theta_{Rvt}V + \hat{V}_R).* |\Xi_{Ru}U + d_{u1}| - (\Theta_{Lvt}V + \hat{V}_L).* |\Xi_{Lu}U + d_{u2}| \right].
\end{aligned}$$

其中 $d_{v1} = \Xi_{Rv2}\tilde{V}_R$, $d_{v2} = \Xi_{Lv2}\tilde{V}_L$, $d_{u1} = \Xi_{Ru2}\tilde{U}_R$, $d_{u2} = \Xi_{Lu2}\tilde{U}_L$, 因为边界向量 \tilde{U}, \tilde{V} 在整个过程中固定, 因此在全阶模型计算中提前在线下步骤中算出 $d_{u_i}, d_{v_i}, i = 1, 2$, 可以一定程度上节约一小部分计算成本。公式中的符号“.”是 MATLAB 语言中的点乘符号, 这里表示两个矩阵对应元素相乘。其它矩阵解释见表 2。

表 2: 矩阵符号说明

矩阵名	用途	矩阵名	用途
M_{Ru}, M_{Lu}	$(U^2)_x$ 项平均值	M_{Rv}, M_{Lv}	$(V^2)_y$ 项平均值
M_{Rut}, M_{Lut}	$(U^2)_x$ 项差分	M_{Rvt}, M_{Lvt}	$(V^2)_y$ 项差分
Θ_{Ru}, Θ_{Lu}	$(UV)_y$ 项中 U 的平均值	Θ_{Rv}, Θ_{Lv}	$(UV)_x$ 项中 V 的平均值
$\Theta_{Rut}, \Theta_{Lut}$	$(UV)_x$ 项中 U 的差分	$\Theta_{Rvt}, \Theta_{Lvt}$	$(UV)_x$ 项中 V 的差分
Ξ_{Rv}, Ξ_{Lv}	$(UV)_x$ 项中 V 的平均值	Ξ_{Rv2}, Ξ_{Lv2}	$(UV)_x$ 项中 V 的平均值
Ξ_{Ru}, Ξ_{Lu}	$(UV)_y$ 项中 U 的平均值	Ξ_{Ru2}, Ξ_{Lu2}	$(UV)_x$ 项中 U 的平均值

接下来, 首先探索平流项和扩散项的降阶模型。为了得到整个 Navier-Stokes 方程的降阶模型, 最重要的是确保速度场 U, V 始终在低维情况下运算, 这里采用 POD 方法获取基向量, 找出 U, V 的近似值:

$$U(t^n) \approx \Psi_u \alpha(t^n), V(t^n) \approx \Psi_v \beta(t^n), \quad (34)$$

其中 $\alpha(t^n) \in \mathbb{R}^{m_1}, \beta(t^n) \in \mathbb{R}^{m_2}$, 并且 $\Psi_u \in \mathbb{R}^{n_{xy} \times m_1}, \Psi_v \in \mathbb{R}^{n_{xy} \times m_2}$ 的每一列都满足两两正交。将公式(34)代入向量 U, V (31), 再分别乘以 Ψ_u^T, Ψ_v^T , 得到更小未知数 $m_1 \ll n_{xy}, m_2 \ll n_{xy}$ 维度的POD 模型。需要注意的是, 在平流项

$$\begin{aligned}
f_1^{POD}(\alpha, \beta) &= \underbrace{\Psi_u^T}_{m_1 \times n_{xy}} \underbrace{f_1(\Psi_u \alpha, \Psi_v \beta)}_{n_{xy} \times 1} \\
f_2^{POD}(\alpha, \beta) &= \underbrace{\Psi_v^T}_{m_2 \times n_{xy}} \underbrace{f_2(\Psi_u \alpha, \Psi_v \beta)}_{n_{xy} \times 1}
\end{aligned}$$

中基于 POD 计算矩阵-向量乘积的计算复杂度取决于完整模型的未知数数量, 这可能导致 POD 模型效率并不高效。

因此接下来, 需要将 DEIM 应用于非线性项(35)。同 POD 基 Ψ_u, Ψ_v 一样, 首先找出在时间 $t_k \in \{t_1, \dots, t_l\} \subset [0, T]$ 上 $f_1(U, V)$ 和 $f_2(U, V)$ 的 snapshots, 对 $f_1(U, V)$ 和 $f_2(U, V)$ 进行 POD/DEIM 方

法降阶处理，根据 DEIM 近似函数(30)，可以得到以下公式：

$$\begin{aligned} f_1^{POD} &\simeq \hat{f}_1^{DEIM} = \underbrace{\Psi_u^T \Phi_{f_1} (S_1^T \Phi_{f_1})^{-1}}_{\Phi f_1 \in \mathbb{R}^{m_1 \times \tau_1}} \underbrace{S_1^T f_1 (\Psi_u \alpha, \Psi_v \beta)}_{\tilde{f}_1(\alpha, \beta) \in \mathbb{R}^{\tau_1 \times 1}}, \\ f_2^{POD} &\simeq \hat{f}_2^{DEIM} = \underbrace{\Psi_v^T \Phi_{f_2} (S_2^T \Phi_{f_2})^{-1}}_{\Phi f_2 \in \mathbb{R}^{m_2 \times \tau_2}} \underbrace{S_2^T f_2 (\Psi_u \alpha, \Psi_v \beta)}_{\tilde{f}_2(\alpha, \beta) \in \mathbb{R}^{\tau_2 \times 1}}. \end{aligned} \quad (36)$$

其中 $\Phi_{f_i} \in \mathbb{R}^{n_{xy} \times \tau_i}, i = 1, 2$ 包含关于最大特征值(或奇异值)的 snapshots $\{f_i(U(t_k), V(t_k)), i = 1, 2; k = 1, 2, \dots, n_k\}$ 张成空间的前 τ_i 个 POD 基。投影插值矩阵 $S_i = [e_{\rho_1}, \dots, e_{\rho_{\tau_i}}] \in \mathbb{R}^{n_{xy} \times \tau_i}, i = 1, 2$ 是通过 DEIM 算法得到的，索引指标 ρ_1, \dots, ρ_i 代表选出 f_i 中的部分行。

需要注意的是参数 γ 在每一个时间步长下都需要通过完整的 U 和 V 计算，这样会增加计算成本。本文做了一个比较简单的处理，在完整模型计算中观察到整个过程中 u, v 的最大值基本固定在小范围内，因此在记录 snapshots 的同时将网格点中记录最大速度点的位置，形成新的指标矩阵 $E_u, E_v = [0, \dots, 0, 1, 0, \dots, 0, 1, 0, \dots, 0]^T$ ，用来选取 POD 基 Ψ_u 和 Ψ_v 中的行向量，从而提高计算速度。 γ 的求解公式变为：

$$\gamma = \min(1.2 \cdot \Delta t \cdot \max(\max|(E_u \cdot \Psi_u)\alpha|, \max|(E_v \cdot \Psi_v)\beta|), 1) \quad (37)$$

这里符号 “.” 代表着内积。由此得出降阶模型的平流项数值求解公式：

$$\begin{aligned} \tilde{f}_1(\alpha, \beta) &:= \frac{1}{4\Delta x} \{ [(S_1^T M_{Ru} \Psi_u \alpha + S_1^T \tilde{U}_R)^2 - (S_1^T M_{Lu} \Psi_u \alpha + S_1^T \tilde{U}_L)^2] \\ &\quad - \gamma [|S_1^T M_{Ru} \Psi_u \alpha + S_1^T \tilde{U}_R|.* (S_1^T M_{Rut} \Psi_u \alpha + S_1^T \tilde{U}_R) - |S_1^T M_{Lu} \Psi_u \alpha + S_1^T \tilde{U}_L|.* (S_1^T M_{Lut} \Psi_u \alpha - S_1^T \tilde{U}_L)] \} \\ &\quad - \frac{1}{4\Delta y} \{ [(S_1^T \Theta_{Ru} \Psi_u \alpha + S_1^T \hat{U}_R).*(S_1^T \Xi_{Rv} \Psi_v \beta + S_1^T d_{v1}) - (S_1^T \Theta_{Lu} \Psi_u \alpha + S_1^T \hat{U}_L).*(S_1^T \Xi_{Lv} \Psi_v \beta + S_1^T d_{v2})] \\ &\quad - \gamma [(S_1^T \Theta_{Rut} \Psi_u \alpha + S_1^T \hat{U}_R).* |S_1^T \Xi_{Rv} \Psi_v \beta + S_1^T d_{v1}| - (S_1^T \Theta_{Lut} \Psi_u \alpha + S_1^T \hat{U}_L).* |S_1^T \Xi_{Lv} \Psi_v \beta + S_1^T d_{v2}|] \}. \quad (38) \\ \tilde{f}_2(\alpha, \beta) &:= \frac{1}{4\Delta y} \{ [(S_2^T M_{Rv} \Psi_v \beta + S_2^T \tilde{V}_R)^2 - (S_2^T M_{Lv} \Psi_v \beta + S_2^T \tilde{V}_L)^2] \\ &\quad - \gamma [|S_2^T M_{Rv} \Psi_v \beta + S_2^T \tilde{V}_R|.* (S_2^T M_{Rvt} \Psi_v \beta + S_2^T \tilde{V}_R) - |S_2^T M_{Lv} \Psi_v \beta + S_2^T \tilde{V}_L|.* (S_2^T M_{Lvt} \Psi_v \beta - S_2^T \tilde{V}_L)] \} \\ &\quad - \frac{1}{4\Delta x} \{ [(S_2^T \Theta_{Rv} \Psi_v \beta + S_2^T \hat{V}_R).*(S_2^T \Xi_{Ru} \Psi_u \alpha + S_2^T d_{u1}) - (S_2^T \Theta_{Lv} \Psi_v \beta + S_2^T \hat{V}_L).*(S_2^T \Xi_{Lu} \Psi_u \alpha + S_2^T d_{u2})] \\ &\quad - \gamma [(S_2^T \Theta_{Rvt} \Psi_v \beta + S_2^T \hat{V}_R).* |S_2^T \Xi_{Ru} \Psi_u \alpha + S_2^T d_{u1}| - (S_2^T \Theta_{Lvt} \Psi_v \beta + S_2^T \hat{V}_L).* |S_2^T \Xi_{Lu} \Psi_u \alpha + S_2^T d_{u2}|] \}. \end{aligned}$$

因为 S_1, S_2, Ψ_u, Ψ_v 固定，不妨设 $\tilde{M}_{Ru} = S_1^T M_{Ru} \Psi_u, \tilde{M}_{Lu} = S_1^T M_{Lu} \Psi_u$ ，并用相同的方法处理公式中剩余也可以提前计算的矩阵乘法。这些矩阵因为在整个计算过程中保持不变，在实际应用中可在线下步骤中算出，不会造成多余的计算复杂度。同时，因为边界向量在整个过程中固定，也可以提前算出 $S_1^T \tilde{U}_R, S_1^T \tilde{U}_L, S_2^T \tilde{V}_R, S_2^T \tilde{V}_L$ ，设

$$\begin{aligned} \tilde{\alpha}_R &= S_1^T \tilde{U}_R, \quad \tilde{\alpha}_L = S_1^T \tilde{U}_L, \quad \tilde{\beta}_R = S_2^T \tilde{V}_R, \quad \tilde{\beta}_L = S_2^T \tilde{V}_L \\ \tilde{d}_{v1} &= S_1^T d_{v1}, \quad \tilde{d}_{v2} = S_1^T d_{v2}, \quad \tilde{d}_{u1} = S_2^T d_{u1}, \quad \tilde{d}_{u2} = S_2^T d_{u2} \end{aligned}$$

此外虚点向量依旧保持在原维度，对此本文依据虚点向量的生成规则：

$$\hat{U}_R = 2\tilde{U}_N - E_N U,$$

这里， \tilde{U}_N 表示北边边界向量，矩阵 E_N 代表筛选矩阵，在这里代表筛选出矩阵 U 中第一行(北)的元素，对虚点向量也进行了降维处理，有

$$S_1^T \hat{U}_R = S_1^T (2\tilde{U}_N - E_N U) = 2S_1^T \tilde{U}_N - S_1^T E_N \Psi_u \alpha.$$

因为边界向量固定，不妨设 $\tilde{\alpha}_N = 2S_1^T \tilde{U}_N$ ，设 $\hat{E}_N = S_1^T E_N \Psi_u$ ，将这些不随时间变化的矩阵提前在线下过程中算出。最终将 \tilde{f}_1, \tilde{f}_2 的计算公式进一步简化为：

$$\begin{aligned}
\tilde{f}_1(\alpha, \beta) &:= \frac{1}{4\Delta x} \left\{ \left[(\dot{M}_{Ru}\alpha + \tilde{\alpha}_R)^2 - (\dot{M}_{Lu}\alpha + \tilde{\alpha}_L)^2 \right] \right. \\
&\quad - \gamma [|\dot{M}_{Ru}\alpha + \tilde{\alpha}_R|.* (\dot{M}_{Rut}\alpha + \tilde{\alpha}_R) - |\dot{M}_{Lu}\alpha + \tilde{\alpha}_L|.* (\dot{M}_{Lut}\alpha - \tilde{\alpha}_L)] \} \\
&\quad - \frac{1}{4\Delta y} \left\{ [(\Theta_{Ru}\alpha + \tilde{\alpha}_N - \hat{E}_N\alpha).* (\dot{\Xi}_{Rv}\beta + \dot{d}_{v1}) - (\Theta_{Lu}\alpha + \tilde{\alpha}_S - \hat{E}_S\alpha).* (\dot{\Xi}_{Lv}\beta + \dot{d}_{v2})] \right. \\
&\quad \left. - \gamma [(\Theta_{Rut}\alpha + \tilde{\alpha}_N - \hat{E}_N\alpha).* |\dot{\Xi}_{Rv}\beta + \dot{d}_{v1}| - (\Theta_{Lut}\alpha + \tilde{\alpha}_S - \hat{E}_S\alpha).* |\dot{\Xi}_{Lv}\beta + \dot{d}_{v2}|] \right\}. \quad (39) \\
\tilde{f}_2(\alpha, \beta) &:= \frac{1}{4\Delta y} \left\{ \left[(\dot{M}_{Rv}\beta + \tilde{\beta}_R)^2 - (\dot{M}_{Lv}\beta + \tilde{\beta}_L)^2 \right] \right. \\
&\quad - \gamma [|\dot{M}_{Rv}\beta + \tilde{\beta}_R|.* (\dot{M}_{Rvt}\beta + \tilde{\beta}_R) - |\dot{M}_{Lv}\beta + \tilde{\beta}_L|.* (\dot{M}_{Lvt}\beta - \tilde{\beta}_L)] \} \\
&\quad - \frac{1}{4\Delta x} \left\{ [(\Theta_{Rv}\beta + \tilde{\beta}_W - \hat{E}_W\beta).* (\dot{\Xi}_{Ru}\alpha + \dot{d}_{u1}) - (\Theta_{Lv}\beta + \tilde{\beta}_E - \hat{E}_E\beta).* (\dot{\Xi}_{Lu}\alpha + \dot{d}_{u2})] \right. \\
&\quad \left. - \gamma [(\Theta_{Rvt}\beta + \tilde{\beta}_W - \hat{E}_W\beta).* |\dot{\Xi}_{Ru}\alpha + \dot{d}_{u1}| - (\Theta_{Lvt}\beta + \tilde{\beta}_E - \hat{E}_E\beta).* |\dot{\Xi}_{Lu}\alpha + \dot{d}_{u2}|] \right\}.
\end{aligned}$$

扩散项 $H_u(U + k_x \tilde{U} + k_y \tilde{V}), H_v(V + k_y \tilde{V} + k_x \tilde{U})$ 在 U, V 降阶的基础上，稍作修正，设 $H_u^{POD} = \Psi_u^T H_u \Psi_u, H_v^{POD} = \Psi_v^T H_v \Psi_v, H_u^{POD} \in \mathbb{R}^{m_1}, H_v^{POD} \in \mathbb{R}^{m_2}$ ，这些可以在线下过程中提前计算。虚点向量与平流项中的处理方式相同，即

$$H_u \hat{U} = H_u^{POD} (2\Psi_u^T \tilde{U}_{NS} - \Psi_u^T E_{NS} \Psi_u \alpha),$$

同理得到 $H_v \hat{U}$ 。在这一步中，也能利用线下的时间得到 $\tilde{\alpha}_{NS} = 2\Psi_u^T \tilde{U}_{NS}, \hat{E}_{NS} = \Psi_u^T E_{NS} \Psi_u$ ，其他矩阵相似处理，得到关于公式(17)的矩阵-向量形式降阶模型

$$\begin{cases} \frac{\alpha^{**} - \alpha^n}{\Delta t} = -\hat{f}_1^{DEIM}(\alpha^n, \beta^n) + H_u^{POD} [\alpha^n + k_x \Psi_u^T \tilde{U} + k_y (\tilde{\alpha}_{NS} + \hat{E}_{NS} \alpha)] \\ \frac{\beta^{**} - \beta^n}{\Delta t} = -\hat{f}_2^{DEIM}(\alpha^n, \beta^n) + H_v^{POD} [\beta^n + k_y \Psi_v^T \tilde{V} + k_x (\tilde{\beta}_{WE} + \hat{E}_{WE} \beta)] \end{cases}. \quad (40)$$

(三) POD/DEIM 方法构造压力梯度项的降阶模型

结束上述的步骤后，得到了低维度的中间速度 α^{**} 和 β^{**} ，接下来需要考虑解 Poisson 方程(20)来更正速度场。根据投射法知由流速场在时间 $n+1$ 上收敛，有 $\nabla U^{n+1} = 0, \nabla V^{n+1} = 0$ ，因此满足

$$\begin{cases} \frac{\nabla \tilde{U}^{**}}{\Delta t} = \frac{\partial^2 P^{n+1}}{\partial x^2} \\ \frac{\nabla \tilde{V}^{**}}{\Delta t} = \frac{\partial^2 P^{n+1}}{\partial y^2} \end{cases} \quad (41)$$

将上式进一步化简为：

$$\Delta P = \frac{1}{dt} [\nabla \cdot (U^{**} \ n \ V^{**})]. \quad (42)$$

为方便降维，转换为矩阵向量形式：

$$L_p P = \frac{1}{dt} [R \Gamma_u (\Psi_u \alpha^{**} + \tilde{U}^{**}) + \Gamma_v (\Psi_v \beta^{**} + \tilde{V}^{**})]. \quad (43)$$

L_p 为诺依曼-泊松矩阵， $U^{**} \approx \Psi_u \alpha^{**}, V^{**} \approx \Psi_v \beta^{**}, \tilde{U}^{**}, \tilde{V}^{**}$ 是 U^{**}, V^{**} 的边界， U^{**}, V^{**} 矩阵分别从横向和纵向展开为向量， R 矩阵实现将 U^{**} 改为从纵向展开的向量。令

$$F = \frac{1}{dt} [R \Gamma_u (\Psi_u \alpha^{**} + \tilde{U}^{**}) + \Gamma_v (\Psi_v \beta^{**} + \tilde{V}^{**})], \quad (44)$$

则有

$$L_p P = F. \quad (45)$$

注意到计算该公式是在完整维度上的，可以考虑用 POD/DEIM 方法提取出向量 P 或者 F 的有效值，获得对应的低阶形式，从而提高计算效率。若对压力项 P 降维，为了得到低阶 POD 模型，采用 POD 基来近似 P ：

$$P = \Psi_p c_p \quad (46)$$

再将此式代入(45)，得到

$$c_p = \Psi_p^T L_p F. \quad (47)$$

从理论上分析，若用(47)还原完整空间的 P 存在数值问题

$$P \neq \Psi_p \Psi_p^T L_p F, \quad \text{for } \Psi_p \Psi_p^T \neq I \quad (48)$$

其中 I 是单位向量。这样处理的后果是每个时间间隔里更正中间速度的步骤都会造成一定的误差，一旦长时间累积会造成较大且不可逆转的错误结果。由此，本文选择对压力梯度项 F 做降阶模型。原理同平流项相似，先在时间 $t_k \in \{t_1, \dots, t_l\} \subset [0, T]$ 上对完整维度 F 取 snapshot，再进行 POD-DEIM 处理。根据 POD 降阶，有 $F \approx \Psi_F c_F$ ，并通过 DEIM 算法求得指标插值矩阵 S_3 ，得到：

$$\hat{F}^{DEIM} \approx \Psi_F (S_3^T \Psi_F)^{-1} S_3^T F. \quad (49)$$

为达到降阶目的，在计算 F 时，先通过与插值矩阵 S_3 相乘得到低阶的 \tilde{F} ，即

$$\tilde{F} = S_3^T F = \frac{1}{dt} [S_3^T R \Gamma_u (\Psi_u \alpha^{**} + \tilde{U}^{**}) + S_3^T \Gamma_v (\Psi_v \beta^{**} + \tilde{V}^{**})] \quad (50)$$

因为 $R, S_3^T, \Gamma_u, \Gamma_v$ 固定，这里设 $\tilde{\Gamma}_u = S_3^T R \Gamma_u \Psi_u, \tilde{\Gamma}_{ub} = S_3^T R \Gamma_u, \tilde{\Gamma}_v = S_3^T \Gamma_v \Psi_v, \tilde{\Gamma}_{vb} = S_3^T \Gamma_v$ ，从而 \tilde{F} 的计算式简化为：

$$\tilde{F} = \frac{1}{dt} [\tilde{\Gamma}_u \alpha^{**} + \tilde{\Gamma}_{ub} \tilde{U}^{**} + \tilde{\Gamma}_v \beta^{**} + \tilde{\Gamma}_{vb} \tilde{V}^{**}]. \quad (51)$$

结合(45)，可以得到 P 的求解公式

$$P = L_p^{-1} \hat{F}^{DEIM} = L_p^{-1} \Psi_F (S_3^T \Psi_F)^{-1} \tilde{F}. \quad (52)$$

根据全阶模型的数值求解，可以获得降维流速场更新公式为：

$$\begin{aligned} \alpha^{n+1} &= \alpha^{**} - dt \Psi_u^T A_1 L_p^{-1} \Psi_F (S_3^T \Psi_F)^{-1} \tilde{F}, \\ \beta^{n+1} &= \beta^{**} - dt \Psi_v^T A_2 L_p^{-1} \Psi_F (S_3^T \Psi_F)^{-1} \tilde{F}. \end{aligned} \quad (53)$$

其中 A_1, A_2 是对 P 不同方向的梯度算子矩阵，最后得到 $\alpha^{n+1}, \beta^{n+1}$ 。基于矩阵 $\Psi_u^T, \Psi_v^T, A_1, A_2, L_p^{-1}, \Psi_F, S_3$ 在时间上固定的特性，不妨设

$$\begin{aligned} B_1 &= \Psi_u^T A_1 L_p^{-1} \Psi_F (S_3^T \Psi_F)^{-1}, \\ B_2 &= \Psi_v^T A_2 L_p^{-1} \Psi_F (S_3^T \Psi_F)^{-1}, \end{aligned}$$

这些矩阵在线下过程中被提前计算，从而公式(53)进一步简写为：

$$\begin{aligned} \alpha^{n+1} &= \alpha^{**} - dt B_1 \tilde{F}, \\ \beta^{n+1} &= \beta^{**} - dt B_2 \tilde{F}. \end{aligned} \quad (54)$$

这样就保证了时间点 n 到 $n+1$ 中流场速度完全在低维空间中计算，即经过降阶模型的平流项和扩散项求出低阶中间速度 α^{**}, β^{**} ，再根据公式(50)算出低维度的 \tilde{F} ，直接带入速度更正公式(54)即可求出下一时间点的低阶速度场。如此一来，Navier-Stokes 方程的完全降阶模型已经全部构建完成。

五、数值实验

在本节中，通过完整模型和 POD/DEIM 方法的降阶模型的数值实验结果，旨在说明 POD/DEIM 方法降阶模型的准确性和效率。这里计算域取为 $\Omega = [0,1] \times [0,1]$ 。在雷诺数 $Re = 100$ 的情况下取不同的网格点和时间步长进行实验对比计算时长和误差。这里将北边界的速度 u 的速度初始化为 1，其余各点速度都初始化为 0，边界条件采用第二章中提及的盖驱动空腔问题的条件。

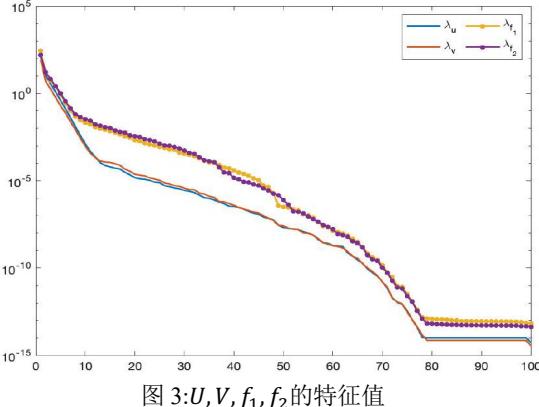


图 3: U, V, f_1, f_2 的特征值

为了评估降阶模型与完整模型的近似程度，采用均方误差(RMSE)测量在时间点 n 上降阶模型与完整模型速度 U 的误差：

$$\text{RMSE}^n = \sqrt{\frac{\sum_{i=1}^N (U_i^n - U_{0,i}^n)^2}{N}}, \quad (55)$$

除此以外，也可以通过使用相对误差 (\mathbf{E}) 来分析误差，其计算公式定义如下：

$$\mathbf{E}_u = \frac{\|U^{Full} - U^{POD/DEIM}\|_2}{\|U^{Full}\|_2} \quad (56)$$

其中上标 *full* 代表着完整维度的模型，POD/DEIM 表示降阶模型。

首先计算等间隔的时间步长下的完整维度 Navier-Stokes 方程数值解，获取 100 个 $U, V, f_1(U, V)$ 和 $f_2(U, V)$ 的 snapshots，然后利用这些 snapshots 构建出的相应的 POD 基。需要注意的是本实验的完整模型也是计算矩阵-向量形式的变量 U, V, P 。图 3 展示了 U, V 和非线性项 f_1, f_2 的 snapshots 解的特征值衰减情况。本次实验 U, V 的 POD 基的维数都取 20，因为这样能够捕获系统中超过 99.9% ($I_u(k) \geq 99.9\%$) 的能量。将 DEIM 方法作用到 f_1 和 f_2 的 POD 基，选择前 20 个空间插值点作为输入，这样能够显著提高 POD 降阶模型的效率并实现非线性项的复杂性降低。 f_1 和 f_2 分布见图 4。同理对 F 进行 POD/DEIM 操作，选取的 POD 维度为 20，DEIM 维度为 30。

表 3 示了 U, V 的 POD 维度取值均为 20，的 DEIM 维度为 20， F 的 POD 和 DEIM 维度分别为 20、30 下，设定时间步长为 0.001，一共计算 2000 步时，不同的网格点数下完整模型和 POD/DEIM 降阶模型的 CPU 运行时长的对比。从表中能看出 POD/DEIM 方法得到的降阶模型明显缩短了 CPU 运行时间。从图 5 能看出在时间步长为 0.01， $t = 2$ 秒时，降阶模型的速度 u 与完整模型是非常接近的，较难肉眼分辨出两种模型的不同。图 5 描绘了速度 u 在空间上不同点的误

差，可以发现靠近北方的速度更接近完整模型，但是误差最大处的误差值仍较小。

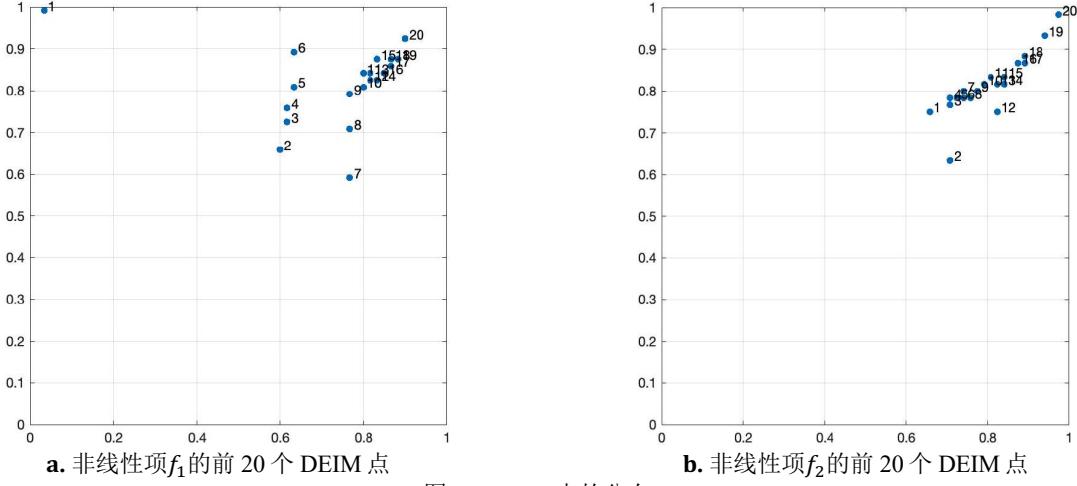


图 4: DEIM 点的分布

表 3: 完整模型和降阶模型的CPU时间对比 (时间单位: 秒)

网格数	完整模型	降阶模型
30×30	1.953725	0.139469
60×60	10.341774	0.081591
90×90	27.639257	0.135371
120×120	56.015332	0.216241
150×150	92.007303	0.765497

此外本文还给出了可视化流场的方式下完整模型和降阶模型的对比。在这里，速度场显示为箭型，即小箭头指示流动方向。因为不同区域的流速变化很大，因此输出归一化速度场，并同时显示流线。注意到由于速度场无发散，因此流线是闭合的。最后得到两种模型在 $t = 2$ 秒时的流场图对比 6。从图 5 可以发现，降阶模型在最后一个时间点上的流场与完整模型仍是较为相似的。

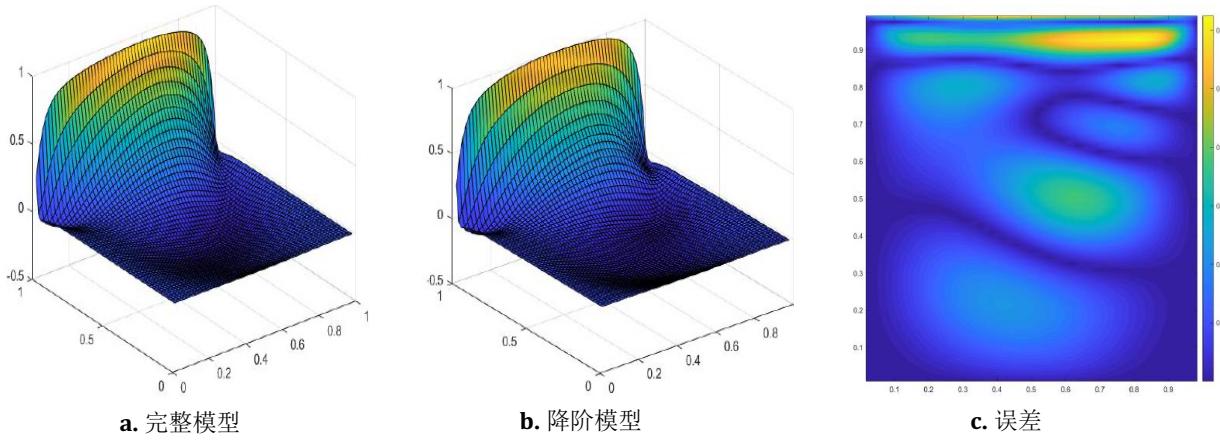


图 5: $t = 2$ s 时完整模型和降阶模型下速度场 u 的对比图

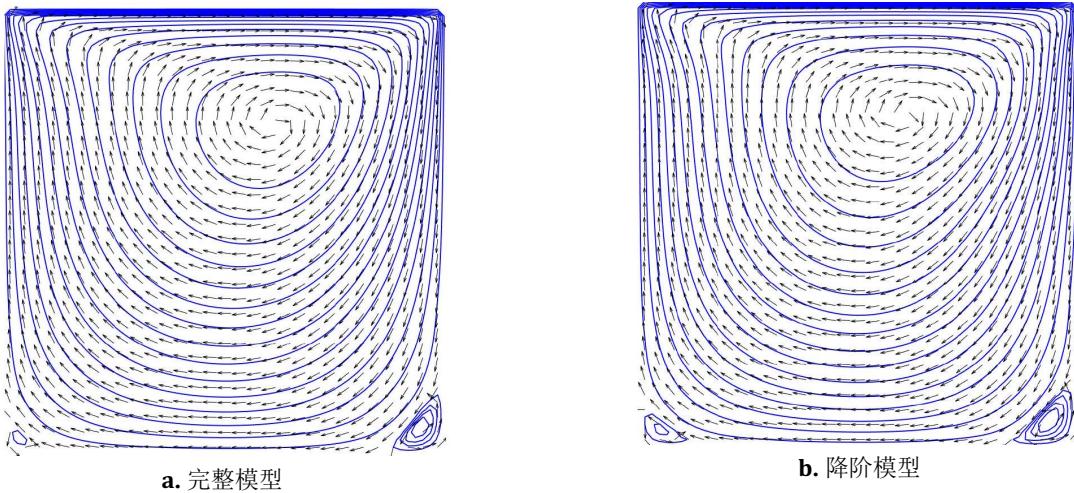


图 6: $t = 2\text{s}$ 时的流场图对比($Re = 100$)

六、总结

本文利用 POD/DEIM 方法构造了有限差分形式的 Navier-Stokes 方程的降阶模型，这种方式比较新颖。为简单明了地阐述结果，本文显示的计算结果只针对速度场 u ，不提供速度场 v 以及压力项 p 的对比结果。雷诺数 $Re = 100$ (代表湍流情况)的情况下，用 POD 方法对速度场 U, V 和压力梯度项 F 降维到 20 阶，这样能捕获超过 99.9% 的系统能量，再分别对 f_1, f_2 选择 20 个 DEIM 插值点，对 F 选择 30 个 DEIM 插值点。DEIM 插值点的选择能显著降低计算速度。在实验中，观察到了从 0 时刻到 2000 个时间步长，降阶模型的 CPU 运行速度显著小于全阶系统的速度，两种模型的速度差也和理论上的时间复杂度之差，分别为 $\mathcal{O}(q_1 m_1^2)$ 和 $\mathcal{O}(96n^2)$ (见附录 B)。

本文在某些细节上的处理仍有不足。如何选出最优的 POD 维度仍是一个未解决的问题，POD 维度小会造成计算精度的降低，而维度增加会导致计算成本也增加，同时文章中仅依靠能量占比原则(26)选择 POD 基不够准确。此外，本文设计到了较多参数的降阶处理，只是对每一个参数分别按照一定规则提取了有效元素，但没有对比哪些参数的维数变化会对模型的速度增加带来更加显著影响，也没有考虑参数彼此之间相关性带来的影响。而对于较大雷诺数的情况，本文并未讨论。雷诺数大可能会造成模型精准度降低，但是可能仅仅通过增加 POD/DEIM 方法的维度并不能明显减小误差。这些将在后续的研究工作中继续进行。

参考文献

- [1] Newton I. Philosophiae naturalis principia mathematica: Londoni, Jussu Societatis Regiae ac typis Josephi Streater, 1687[M]. W. Dawson, 1953.
- [2] Navier C. Mémoire sur les lois du mouvement des fluides[J]. Mémoires de l'Académie Royale des Sciences de l'Institut de France, 1823, 6(1823): 389-440.
- [3] Poisson S D. Mémoire sur les équations générales de l'équilibre et du mouvement des corps solides élastiques et de fluides[M]. L'imprimerie Royale, 1831.

- [4] de Saint-Venant B. Notea joindre au mémoire sur la dynamique des fluides[J]. Comptes rendus, 1843, 17(1240): 1843.
- [5] Stokes G G. On the theories of the internal friction of fluids in motion, and of the equilibrium and motion of elastic solids[J]. 2007.
- [6] McLean D. Understanding aerodynamics: arguing from the real physics[M]. John Wiley & Sons, 2012.
- [7] Giraldo F X, Restelli M, Lauter M. Semi-implicit formulations of the Navier–Stokes equations: application to nonhydrostatic atmospheric modeling[J]. SIAM Journal on Scientific Computing, 2010, 32(6): 3394-3425.
- [8] Marshall J, Adcroft A, Hill C, et al. A finite-volume, incompressible Navier Stokes model for studies of the ocean on parallel computers[J]. Journal of Geophysical Research: Oceans, 1997, 102(C3): 5753-5766.
- [9] Wille S Ø. Adaptive finite element simulations of the surface currents in the North Sea[J]. Computer methods in applied mechanics and engineering, 1998, 166(3-4): 379-390.
- [10] Willis A P. The Openpipeflow Navier–Stokes solver[J]. SoftwareX, 2017, 6: 124-127.
- [11] Kaliakatsos C, Tsangaris S. Motion of deformable drops in pipes and channels using Navier–Stokes equations[J]. International journal for numerical methods in fluids, 2000, 34(7): 609-626.
- [12] Khairani C, Marpaung T J. Computational analysis of fluid behaviour around airfoil with Navier-Stokes equation[C]. Journal of Physics: Conference Series. IOP Publishing, 2019, 1376(1): 012003.
- [13] Pivkin I V, Hueso E, Weinstein R, et al. Simulation and visualization of air flow around bat wings during flight[C]. Computational Science–ICCS 2005: 5th International Conference, Atlanta, GA, USA, May 22-25, 2005. Proceedings, Part II 5. Springer Berlin Heidelberg, 2005: 689-694.
- [14] Chanyal B C. Quaternionic approach on the Dirac–Maxwell, Bernoulli and Navier–Stokes equations for dyonic fluid plasma[J]. International Journal of Modern Physics A, 2019, 34(31): 1950202.
- [15] Glowinski R, Pironneau O. Finite element methods for Navier-Stokes equations[J]. Annual review of fluid mechanics, 1992, 24(1): 167-204.
- [16] Girault V, Raviart P A. Finite element methods for Navier-Stokes equations: theory and algorithms[M]. Springer Science & Business Media, 2012.
- [17] Taylor C, Hood P. A numerical solution of the Navier-Stokes equations using the finite element technique[J]. Computers & Fluids, 1973, 1(1): 73-100.
- [18] Franca L P, Frey S L. Stabilized finite element methods: II. The incompressible Navier-Stokes equations[J]. Computer Methods in Applied Mechanics and Engineering, 1992, 99(2-3): 209-233.
- [19] Mu L, Ye X. A simple finite element method for the Stokes equations[J]. Advances in Computational Mathematics, 2017, 43: 1305-1324.
- [20] Begue C, Conca C, Murat F, et al. Les équations de Stokes et de Navier-Stokes avec des conditions aux limites sur la pression[C]. Nonlinear partial differential equations and their applications, College de France Seminar. Pitman, 1988, 9: 179-264.
- [21] Harlow F H, Welch J E. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface[J]. The physics of fluids, 1965, 8(12): 2182-2189.
- [22] Dogan A. A Galerkin finite element approach to Burgers' equation[J]. Applied mathematics and computation, 2004, 157(2): 331-346.
- [23] Aksan E N. A numerical solution of Burgers' equation by finite element method constructed on the method of discretization in time[J]. Applied mathematics and computation, 2005, 170(2): 895-904.
- [24] Arminjon P, Beauchamp C. Numerical solution of Burgers' equations in two space dimensions[J]. Computer Methods in Applied Mechanics and Engineering, 1979, 19(3): 351-365.
- [25] Basto M, Semiao V, Calheiros F. Dynamics and synchronization of numerical solutions of the Burgers equation[J]. Journal of computational and applied mathematics, 2009, 231(2): 793-806.
- [26] Bahadir A R. A fully implicit finite-difference scheme for two-dimensional Burgers' equations[J]. Applied Mathematics and Computation, 2003, 137(1): 131-137.
- [27] Fletcher C A J. A comparison of finite element and finite difference solutions of the one-and two-dimensional Burgers' equations[J]. Journal of Computational Physics, 1983, 51(1): 159-188.
- [28] Chorin A J. Numerical solution of the Navier-Stokes equations[J]. Mathematics of computation, 1968, 22(104): 745-762.
- [29] Kim J, Moin P. Application of a fractional-step method to incompressible Navier-Stokes equations[J]. Journal of

- computational physics, 1985, 59(2): 308-323.
- [30] Temam R. Sur l'approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionnaires (II)[J]. Archive for rational mechanics and analysis, 1969, 33: 377-385.
 - [31] Van Kan J. A second-order accurate pressure-correction scheme for viscous incompressible flow[J]. SIAM journal on scientific and statistical computing, 1986, 7(3): 870-891.
 - [32] Guermond J L, Shen J. Quelques résultats nouveaux sur les méthodes de projection[J]. Comptes Rendus de l'Académie des Sciences-Series I-Mathematics, 2001, 333(12): 1111-1116.
 - [33] Guermond J L, Shen J. Velocity-correction projection methods for incompressible flows[J]. SIAM Journal on Numerical Analysis, 2003, 41(1): 112-134.
 - [34] Karniadakis G E, Israeli M, Orszag S A. High-order splitting methods for the incompressible Navier-Stokes equations[J]. Journal of computational physics, 1991, 97(2): 414-443.
 - [35] Orszag S A, Israeli M, Deville M O. Boundary conditions for incompressible flows[J]. Journal of Scientific Computing, 1986, 1: 75-111.
 - [36] Feng L. Review of model order reduction methods for numerical simulation of nonlinear circuits[J]. Applied Mathematics and Computation, 2005, 167(1): 576-591.
 - [37] Yvonnet J, He Q C. The reduced model multiscale method (R3M) for the non-linear homogenization of hyperelastic media at finite strains[J]. Journal of Computational Physics, 2007, 223(1): 341-368.
 - [38] Maier C, Epureanu A, Marinescu V, et al. A new concept of the reduced order modeling in metal forming[C]. Proceedings of the 6th WSEAS international conference on Dynamical systems and control, CONTROL. 2010, 10: 133-136.
 - [39] Akhtar I, Borggaard J, Hay A. Shape sensitivity analysis in flow models using a finite-difference approach[J]. Mathematical Problems in Engineering, 2010, 2010.
 - [40] Bergmann M, Cordier L, Bracher J P. Drag minimization of the cylinder wake by trust-region proper orthogonal decomposition[C]. Active Flow Control: Papers contributed to the Conference “Active Flow Control 2006”, Berlin, Germany, September 27 to 29, 2006. Springer Berlin Heidelberg, 2007: 309-324.
 - [41] Lassila T, Rozza G. Parametric free-form shape design with PDE models and reduced basis method[J]. Computer Methods in Applied Mechanics and Engineering, 2010, 199(23-24): 1583-1592.
 - [42] Rozza G, Lassila T, Manzoni A. Reduced basis approximation for shape optimization in thermal flows with a parametrized polynomial geometric map[C]. Spectral and High Order Methods for Partial Differential Equations: Selected papers from the ICOSAHOM'09 conference, June 22-26, Trondheim, Norway. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010: 307-315.
 - [43] Arian E, Fahl M, Sachs E W. Trust-region proper orthogonal decomposition for flow control[R]. INSTITUTE FOR COMPUTER APPLICATIONS IN SCIENCE AND ENGINEERING HAMPTON VA, 2000.
 - [44] Reduced-order modelling for flow control[M]. Springer Science & Business Media, 2011.
 - [45] Graham W R, Peraire J, Tang K Y. Optimal control of vortex shedding using low-order models. Part I—open-loop model development[J]. International Journal for Numerical Methods in Engineering, 1999, 44(7): 945-972.
 - [46] Baiges J, Codina R, Idelsohn S. A domain decomposition strategy for reduced order models. Application to the incompressible Navier–Stokes equations[J]. Computer Methods in Applied Mechanics and Engineering, 2013, 267: 23-42.
 - [47] Veroy K, Patera A T. Certified real-time solution of the parametrized steady incompressible Navier–Stokes equations: rigorous reduced-basis a posteriori error bounds[J]. International Journal for Numerical Methods in Fluids, 2005, 47(8-9): 773-788.
 - [48] Burkardt J, Gunzburger M, Lee H C. POD and CVT-based reduced-order modeling of Navier–Stokes flows[J]. Computer methods in applied mechanics and engineering, 2006, 196(1-3): 337-355.
 - [49] Ballarin F, Manzoni A, Quarteroni A, et al. Supremizer stabilization of POD–Galerkin approximation of parametrized steady incompressible Navier–Stokes equations[J]. International Journal for Numerical Methods in Engineering, 2015, 102(5): 1136-1161.
 - [50] Xiao D, Fang F, Du J, et al. Non-linear Petrov–Galerkin methods for reduced order modelling of the Navier–Stokes equations using a mixed finite element pair[J]. Computer Methods In Applied Mechanics and Engineering, 2013, 255: 147-157.

- [51] Stabile G, Rozza G. Finite volume POD-Galerkin stabilised reduced order methods for the parametrised incompressible Navier–Stokes equations[J]. *Computers & Fluids*, 2018, 173: 273-284.
- [52] Girfoglio M, Quaini A, Rozza G. A POD-Galerkin reduced order model for a LES filtering approach[J]. *Journal of Computational Physics*, 2021, 436: 110260.
- [53] Lorenzi S, Cammi A, Luzzi L, et al. POD-Galerkin method for finite volume approximation of Navier–Stokes and RANS equations[J]. *Computer Methods in Applied Mechanics and Engineering*, 2016, 311: 151-179.
- [54] Xiao D, Fang F, Buchan A G, et al. Non-linear model reduction for the Navier–Stokes equations using residual DEIM method[J]. *Journal of Computational Physics*, 2014, 263: 1-18.
- [55] Xiao D, Fang F, Pain C, et al. Non-intrusive reduced-order modelling of the Navier–Stokes equations based on RBF interpolation[J]. *International Journal for Numerical Methods in Fluids*, 2015, 79(11): 580-595.
- [56] Pearson K. LIII. On lines and planes of closest fit to systems of points in space[J]. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 1901, 2(11): 559-572.
- [57] Ito K, Ravindran S S. A reduced basis method for control problems governed by PDEs[R]. North Carolina State University. Center for Research in Scientific Computation, 1997.
- [58] Broomhead D S, King G P. Extracting qualitative dynamics from experimental data[J]. *Physica D: Nonlinear Phenomena*, 1986, 20(2-3): 217-236.
- [59] Chaturantabut S. Dimension reduction for unsteady nonlinear partial differential equations via empirical interpolation methods[M]. Rice university, 2009.
- [60] Chaturantabut S, Sorensen D C. Nonlinear model reduction via discrete empirical interpolation[J]. *SIAM Journal on Scientific Computing*, 2010, 32(5): 2737-2764.
- [61] Barrault M, Maday Y, Nguyen N C, et al. An ‘empirical interpolation’method: application to efficient reduced-basis discretization of partial differential equations[J]. *Comptes Rendus Mathematique*, 2004, 339(9): 667-672.
- [62] řtefanescu R, Navon I M. POD/DEIM nonlinear model order reduction of an ADI implicit shallow water equations model[J]. *Journal of Computational Physics*, 2013, 237: 95-114.
- [63] Chaturantabut S, Sorensen D C. A state space error estimate for POD-DEIM nonlinear model reduction[J]. *SIAM Journal on numerical analysis*, 2012, 50(1): 46-63.
- [64] Rozza G, Huynh D B P, Patera A T. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: application to transport and continuum mechanics[J]. *Archives of Computational Methods in Engineering*, 2008, 15(3): 229.
- [65] Boyaval S, Le Bris C, Lelievre T, et al. Reduced basis techniques for stochastic problems[J]. *Archives of Computational methods in Engineering*, 2010, 17: 435-454.
- [66] Eftang J L, Knezevic D J, Patera A T. An hp certified reduced basis method for parametrized parabolic partial differential equations[J]. *Mathematical and Computer Modelling of Dynamical Systems*, 2011, 17(4): 395-422.
- [67] Rozza G, Manzoni A. Model order reduction by geometrical parametrization for shape optimization in computational fluid dynamics[C]. *Proceedings of the ECCOMAS CFD 2010, V European Conference on Computational Fluid Dynamics. 2010 (CONF)*.
- [68] Manzoni A, Quarteroni A, Rozza G. Shape optimization for viscous flows by reduced basis methods and free-form deformation[J]. *International Journal for Numerical Methods in Fluids*, 2012, 70(5): 646-670.
- [69] Negri F, Rozza G, Manzoni A, et al. Reduced basis method for parametrized elliptic optimal control problems[J]. *SIAM Journal on Scientific Computing*, 2013, 35(5): A2316-A2340.
- [70] Wang Y, Navon I M, Wang X, et al. 2D Burgers equation with large Reynolds number using POD/DEIM and calibration[J]. *International Journal for Numerical Methods in Fluids*, 2016, 82(12): 909-931.
- [71] Chorin A J, Marsden J E, Marsden J E. A mathematical introduction to fluid mechanics[M]. New York: Springer, 1990.
- [72] Landau L D, Lifshitz E M. Fluid Mechanics: Landau and Lifshitz: Course of Theoretical Physics, Volume 6[M]. Elsevier, 2013.
- [73] Zienkiewicz O C, Taylor R L, Nithiarasu P. The finite element method for fluid dynamics[M]. Butterworth-Heinemann, 2013.
- [74] Seibold B. A compact and fast Matlab code solving the incompressible Navier-Stokes equations on rectangular domains mit18086 navierstokes. m[J]. Massachusetts Institute of Technology, 2008.

- [75] Chella M A, Collados X R, Bihs H, et al. Numerical and experimental investigation of breaking wave interaction with a vertical slender cylinder[J]. Energy Procedia, 2016, 94: 443-451.

致谢

2023 年的夏季，我的本科生活即将结束。回顾过往，有过喜悦、悲伤、满足、疲惫、甜蜜和苦涩的不同瞬间，众多情感在回忆中交织在一起编织出精彩的大学时光。一路以来，有太多人给予我善意与关爱，在此，我要对这四年里所有关心、帮助我的人表示最诚挚的感谢。

首先感谢我的导师——王曰朋老师。

另外，我要感谢我的家人，感谢他们对我求学路上的辛勤付出和不懈支持。父母的包容有了现在的我，敢冲敢闯，22 岁仍能追寻自己的梦。

附录 A DEIM 算法

Algorithm 1 DEIM

Input: $\{\mathbf{u}_\ell\}_\ell^m \subset \mathbb{R}^n$ linearly independent

Output: $\vec{\varphi} = [\varphi_1, \dots, \varphi_m]^T \in \mathbb{R}^m$

- 1: $[\|\rho\|, \varphi_1] = \max\{|\mathbf{u}_1|\}$
 - 2: $\mathbf{U} = [\mathbf{u}_1], \mathbf{P} = [\mathbf{e}_{\varphi_1}], \vec{\varphi} = [\varphi_1]$
 - 3: **for** $\ell = 2$ to m **do**
 - 4: Solve $(\mathbf{P}^T \mathbf{U} \mathbf{c} = \mathbf{P}^T \mathbf{u}_\ell)$ for \mathbf{c}
 - 5: $\mathbf{r} = \mathbf{u}_\ell - \mathbf{U} \mathbf{c}$
 - 6: $[\|\rho\|, \varphi_\ell] = \max\{|\mathbf{r}|\}$
 - 7: $\mathbf{U} \leftarrow [\mathbf{U} \quad \mathbf{u}_\ell], \mathbf{P} = [\mathbf{P} \quad \mathbf{e}_{\varphi_\ell}], \vec{\varphi} = [\vec{\varphi} \quad \varphi_\ell]$
 - 8: **end for**
-

DEIM 算法中的符号 \max 与 MATLAB 中的函数 \max 相同。因此， $[\|\rho\|, \varphi_\ell] = \max\{|\mathbf{r}|\}$ 意味着 $|\rho| = |r_{\varphi_\ell}| = \max_{i=1, \dots, n}\{|r_i|\}$ ，在相同情况下的情况下采用最小指标。请注意，在每次迭代 $\ell = 1, \dots, m$ 中会重新定义 $\rho = r_{\varphi_\ell}$ 。

附录 B 时间复杂度

本文采用大 O 表示法来评估模型的时间复杂度。

附录 B.1 完整模型

设网格数为 $n \times n$ ，则向量 $U, V, \hat{U}, \hat{V}, \tilde{U}, \tilde{V}$ 的维数均为 $n(n-1) \times 1$ ，下面分别从平流项、扩散项和压力项计算从时间步长 t 到 $t+1$ 的复杂度。

平流项公式(33):

- $M_{Ru}, M_{Lu}, M_{Rut}, M_{Lut}$ 是稀疏矩阵，每个矩阵里面都含有 $2n^2 - 3n$ 个元素；
 $\Theta_{Ru}, \Theta_{Lu}, \Theta_{Rut}, \Theta_{Lut}$ 是含有 $2n^2 - 3n + 1$ 个元素的稀疏矩阵； Ξ_{Ru}, Ξ_{Lu} 也为稀疏矩阵，每个矩阵含有 $2(n-1)^2$ 个元素，则 $M_{Ru}U, M_{Lu}U, M_{Rut}U, M_{Lut}U, \Theta_{Ru}U, \Theta_{Lu}U, \Theta_{Rut}U, \Theta_{Lut}U, \Xi_{Ru}V$ ，以及 $\Xi_{Lu}V$ 的总时间复杂度为 $28n^2 - 46n + 12$
- d_{v1}, d_{u2} 固定不变，在线下过程算出；
- 向量点乘共有 8 次，共计时间复杂度为 $8n(n-1)$ ；

故计算 $f_1(U, V)$ 的时间复杂度为 $36n^2 - 54n + 12$ ，同理得 $f_2(U, V)$ 的复杂度也为 $36n^2 - 54n + 12$ 。

扩散项 $H_u(U + k_x \tilde{U} + k_y \hat{U}), H_v(V + k_x \tilde{V} + k_y \hat{V})$:

- H_u 为稀疏矩阵，共有 $5n^2 - 9n + 2$ 个元素，这里 H_u 与三个向量之和相乘，因此时间复杂度为 $5n^2 - 9n + 2$

同理 $H_v(V + k_x \tilde{V} + k_y \hat{V})$ 的复杂度也为 $5n^2 - 9n + 2$ 。

压力项:

- 对应低维度 F 的计算公式(44), 在完整维度下计算公式为 $F = \frac{1}{dt} [R\Gamma_u(U + \tilde{U}) + \Gamma_v(V + \tilde{V})]$, 其中 $R\Gamma_u, \Gamma_v$ 为稀疏矩阵, 含有 $2(n - 1)^2$ 个元素, 因此这一步的时间复杂度为 $8n^2 - 16n + 8$;
- 完整维度下的 U 更新公式为 $U^{t+1} = U^{**} - dtA_1L_p^{-1}F$, 因为 A_1, L_p^{-1} 固定不变, 先在线下得到乘积结果, 两者的乘积是一个满秩矩阵, 从而更新步骤的时间复杂度为 n^2
- 同理更新 V 的时间复杂度也为 n^2

故处理压力项的时间复杂度为 $10n^2 - 16n + 8$

综上所述, 完整维度下, 总时间复杂度为 $92n^2 - 142n + 28$

附录 B.2 降阶模型

网格数为 $n \times n$ 的情况下设向量 U, V 取 POD 维数为 q_1, f_1, f_2 的 DEIM 维数取 m_1, F 向量的 POD 和 DEIM 的维数分别为 q_2, m_2 , 计算参数 γ 用到的指标向量分别选取了 Ψ_u, Ψ_v 的 s_1, s_2 行, 并且满足 $q_1, q_2, s_1, s_2, m_1, m_2 \ll n^2$ 。下面分别从平流项、扩散项和压力项计算从时间步长 t 到 $t + 1$ 的复杂度。

平流项公式(36):

因有 $\hat{M}_{Ru} = S_1^T M_{Ru} \Psi_u$, 因此 $\hat{M}_{Ru} \in \mathbb{R}^{m_1 \times q_1}$, 同理 $\hat{\Theta}_{Ru}, \hat{\Theta}_{Lu}, \hat{\Theta}_{Rut}, \hat{\Theta}_{Lut}, \hat{\Xi}_{Rv}, \hat{\Xi}_{Lv}$ 的维度均为 $m_1 \times q_1$, 从而计算 $\hat{M}_{Ru}\alpha, \hat{M}_{Lu}\alpha, \hat{M}_{Rut}\alpha, \hat{M}_{Lut}\alpha, \hat{\Theta}_{Ru}\alpha, \hat{\Theta}_{Lu}\alpha, \hat{\Theta}_{Rut}\alpha, \hat{\Theta}_{Lut}\alpha, \hat{\Xi}_{Rv}\beta, \hat{\Xi}_{Lv}\beta$ 的时间复杂度均为 $q_1 m_1$:

- 其中 \hat{E}_N, \hat{E}_S 为 $m_1 \times a$ 维度的矩阵, 故计算 $\hat{E}_N\alpha, \hat{E}_S\alpha$ 的时间复杂度为 $q_1 m_1$;
- $\Phi^{f_1} = \Psi_u^T \Phi_{f_1} (S_1^T \Phi_{f_1})^{-1} \in \mathbb{R}^{q_1 \times m_1}$ 可以提前在线下算出, $\Phi^{f_1} \tilde{f}_1$ 的复杂度为 $q_1 m_1^2$
- 向量点乘共有 8 次, 共计时间复杂度为 $8m_1$;

故计算 $\hat{f}_1^{DEIM}(\alpha, \beta)$ 的时间复杂度为 $q_1 m_1^2 + 12q_1 m_1 + 8m_1$, 同理得 $\hat{f}_2^{DEIM}(\alpha, \beta)$ 的复杂度也为 $q_1 m_1^2 + 12q_1 m_1 + 8m_1$ 。

扩散项公式(40):

- 因为边界项固定, $k_{\Psi_u} \tilde{U}, \tilde{\alpha}_{NS}$ 可以提前在线下步骤算出。除此以外, \hat{E}_{NS} 是 $q_1 \times q_1$ 维度的矩阵, 故 $\hat{E}_{NS}\alpha$ 这一步时间复杂度为 q_1^2 ;
 - H_u^{POD} 的维度为 $q_1 \times q_1$, 故 H_u^{POD} 乘以四个向量之和这一步时间复杂度为 q_1^2 ;
- 故扩散项的总复杂度为 $4q_1^2$ 。

压力项:

根据降阶模型 \hat{F} 的计算公式(51), 其中 $\hat{\Gamma}_u, \hat{\Gamma}_v$ 维度为 $m_2 \times q_2, \hat{\Gamma}_{ub}, \hat{\Gamma}_{vb}$ 则为含有 $2m_2$ 个元素的 $m_2 \times n^2$ 维的稀疏矩阵, 因此 \hat{F} 的计算时间复杂度为 $2q_2 m_2 + 4m_2$:

- 因为 $B_1 \in \mathbb{R}^{q_2 \times m_2}$, 故 $B_1 \hat{F}$ 的时间复杂度为 $q_2 m_2$

故压力项的总复杂度为 $4(q_2 + 1)m_2$

此外，计算 $\gamma = \min(1.2 \cdot \Delta t \cdot \max(\max|E_u \Psi_u \alpha|, \max|E_v \Psi_v \beta|), 1)$ 的时间复杂度为 $s_1 + s_2$ 。综上所述，降阶模型下，总时间复杂度为 $2q_1 m_1^2 + 4q_1^2 + 4(q_2 + 1)m_2 + (24q_1 + 16)m_1 + s_1 + s_2$ 。